

CLAIMS

1. A computer network comprising: - at least two processing nodes each having a processor on which one or more user processes are executed and a respective network interface; and a switching network which operatively connects the at least two processing nodes together, each network interface including a command processor and a memory wherein the command processor of said network interface is configured to allocate exclusively to a user process being executed on the processor with which the network interface is associated one or more segments of addressable memory in said network interface memory as a respective one or more command queues.
2. A computer network as claimed in claim 1, wherein each one of said memory segments allocated as command queues is a contiguous series of memory addresses.
3. A computer network as claimed in claim 1, wherein each memory segment of the network interface memory allocated as a command queue has associated with it a queue descriptor which includes a user process identification.
4. A computer network as claimed in claim 3, wherein each queue descriptor includes an insert pointer identifying within the allocated memory segment the memory address where new commands from the relevant user process are to be written.
5. A computer network as claimed in claim 3, wherein each queue descriptor includes a completed pointer identifying within the allocated memory segment the memory address of the end of the most recent completed command.
6. A computer network as claimed in claim 1, wherein the network interface includes a scheduler configured to identify and schedule any active command queues in the network interface memory.

7. A computer network as claimed in claim 6, wherein the scheduler has two or more run queues with at least one of the run queues being denominated a high priority run queue and at least one other of the run queues being denominated a low priority queue.

5

8. A computer network as claimed in claim 1, wherein said network interface includes at least one of the following resources: a thread processor, short transaction engine and a DMA engine, and each command queue stored in said network interface memory has associated with it a corresponding one or more virtual resources.

10

9. A network interface comprising a command processor and a memory wherein the command processor of said network interface is configured to allocate exclusively to a user process being executed on a processor with which the network interface is associated, one or more segments of addressable memory in said network interface memory as a respective one or more command queues.

15

10. A network interface as claimed in claim 9, wherein each one of said memory segments allocated as command queues is a contiguous series of memory addresses.

20

11. A network interface as claimed either in claim 9, wherein each memory segment of the network interface memory allocated as a command queue has associated with it a queue descriptor which includes a user process identification.

25

12. A network interface as claimed in claim 11, wherein each queue descriptor includes an insert pointer identifying within the allocated memory segment the memory address where new commands from the relevant user process are to be written.

30

13. A network interface as claimed in claim 11, wherein each queue descriptor includes a completed pointer identifying within the allocated memory segment the memory address of the end of the most recent completed command.

14. A network interface as claimed in claim 9, wherein the network interface includes a scheduler configured to identify and schedule any active command queues in the network interface memory.

5 15. A network interface as claimed in claim 14, wherein the scheduler has two or more run queues with at least one of the run queues being denominated a high priority run queue and at least one other of the run queues being denominated a low priority queue.

10 16. A network interface as claimed in claim 9, wherein said network interface includes at least one of the following resources: a thread processor, short transaction engine and a DMA engine, and each command queue stored in said network interface memory has associated with it a corresponding one or more virtual resources.

15

17. A method of storing and running commands issued by a processor having associated with it a network interface comprising a command processor and a network interface memory, comprising the steps of:

20 the network interface receiving a request for a command queue from a user process being executed on the processor;

in response to the request allocating exclusively to the user process a memory segment of the network interface memory as a command queue;

storing one or more commands associated with the user process in said command queue; and

25 running said commands in said command queue without further intervention from said processor.

30 18. A method as claimed in claim 17, wherein in response to requests from a plurality of user processes being executed on the processor, a respective plurality of memory segments of the network interface memory are allocated by the network processor.

19. A method as claimed in claim 17, further comprising the step of said network processor generating a queue descriptor, which includes a user process

identification, for each allocated memory segment.

20. A method as claimed in claim 17, wherein said network interface includes a scheduler and said method further comprises the step of generating a run queue of
5 active command queues in said network interface memory each active command queue containing at least one command awaiting execution.

21. A method as claimed in claim 20, wherein at least two run queues are
10 generated including a high priority run queue and a low priority run queue.